Глава 2

Концептуални модели

Обекти и отношения

Това предварително запознаване с принципите за проектиране на БД.

Обекти (entities)

Всеки конкретен или абстрактен елемент от една информационна система се нарича **обект (entity)**. Всеки обект притежава **идентичност**, която го прави различим от останалите.

Примери: торба с разноцветни топчета, цветовете като абстрактни обекти.

Всеки обект е означен със собствено **име на обекта (идентификатор),** което го прави уникален.

Обектите от едно естество се групират в **класове(множества) обекти**: всички студенти, всички автомобили, всички българи.

Всеки клас има уникално име, което го отличава от другите класове. Множествата могат да бъдат дефинирани по различни начини:

- Множество, дефинирано чрез изброяване (мъжки, женски).
- С помощта на декартово произведение, ако X, Y и Z са множества, X*Y*Z ще бъде множество с елементи от вида (x,y,z), където $x \in X$, $y \in Y$, $z \in Z$.
- С помощта на операциите обединение ($\cup$), пресичане ($\cap$) и разлика (–).

Отношение (релация, връзка)

Едно отношение е бинарна релация между две не непременно различни множества.

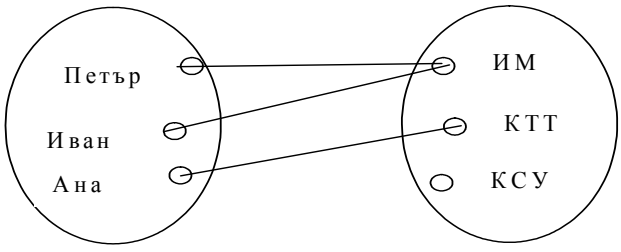
X F/G Y

Където F и G са 2 обратни една на друга функции (по принцип многозначни).

СТУДЕНТ е записан в/студент на СПЕЦИАЛНОСТ

Характеристики на едно отношение

- Еднозначно или многозначно
- Частично или глобално(задължително)
- Минимална и максимална кардиналност



Фигура 2.1 Отношение

Пример:

Класове обекти :

1. СТУДЕНТ (StNo,..) – множеството от всички студенти, които следват в момента или са следвали през последните 10 години.
2. ПРЕПОДАВАТЕЛ
3. СПЕЦИАЛНОСТ
4. ДИСЦИПЛИНА
5. ЗАЛА
6. АДРЕС
7. ИМЕ
8. КУРС (учебен)
9. ДЛЪЖНОСТ {професор, доцент, асистент итн.}

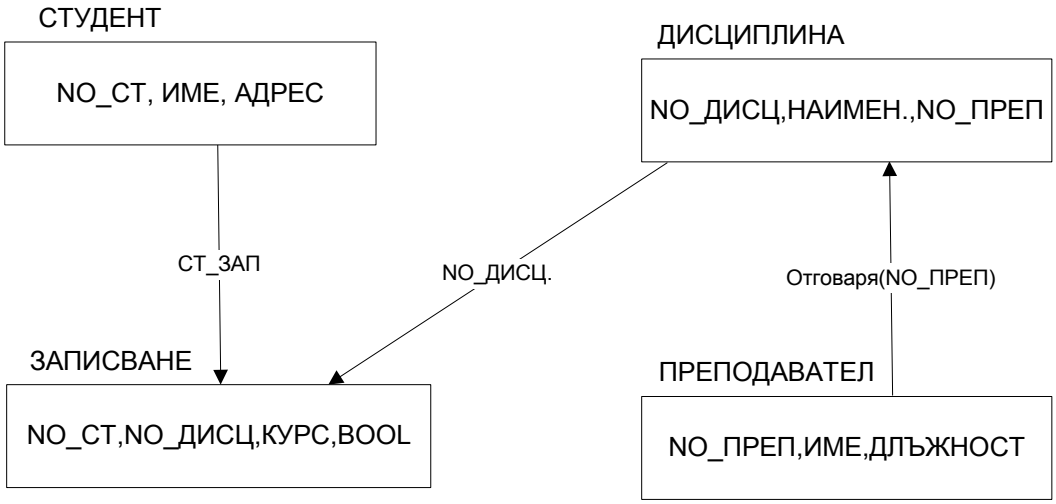
10. ЧАС
11. ДЕН = {понеделник, вторник,…}
12. ЧИСЛО
13. СГРАДА
14. BOOL = {истина, неистина}

Отношения :

15. СТУДЕНТ (1,1) се казва / е името на (0,n) ИМЕ
16. СТУДЕНТ (1,1) живее на / е обитаван от (0,n) АДРЕС
17. СПЕЦИАЛНОСТ (1,n) съдържа / е в (1,n) ДИСЦИПЛИНА
18. ЗАЛА (1,1) има капацитет / е капацитет на (0,n) ЧИСЛО
19. ЗАЛА (1,1) се намира в / съдържа (1,n) СГРАДА
20. СТУДЕНТ (1,1) е записан в / има за студент (0,n) СПЕЦИАЛНОСТ
21. ПРЕПОДАВАТЕЛ (0,n) отговаря за / се води от (1,1) СПЕЦИАЛНОСТ
22. СТУДЕНТ (1,1) следва(л) /е преминат от СПЕЦИАЛНОСТ * КУРС * BOOL
23. СПЕЦИАЛНОСТ (0,n) изисква / е изисквана от (0,n) СПЕЦИАЛНОСТ
24. ДИСЦИПЛИНА (1,n) има за разписание / (0,1) ЧАС * ДЕН * ЗАЛА * СПЕЦИАЛНОСТ
25. ПРЕПОДАВАТЕЛ (1,n) е зает / (0,1) ЧАС * ДЕН * ЗАЛА * ДИСЦИПЛИНА
26. ЗАЛА (1,1) е използвана / (0,1) ЧАС * ДЕН * ДИСЦИПЛИНА * ПРЕПОДАВАТЕЛ
27. СТУДЕНТ (1,1) е завършил/ (0,n) BOOL
28. $[(h,j,s,e) \in$ е зает$(u)] \equiv [(h,j,s,u) \in$ зает$(e)]$
    $h \in$ ЧАС, $j \in$ ДЕН, $s \in$ ЗАЛА, $e \in$ ПРЕПОДАВАТЕЛ, $u \in$ ДИСЦИПЛИНА
29. $[(h,j,u) \in$ използван$(s)] \Rightarrow [\exists\ e \in$ ПРЕПОДАВАТЕЛ $\bullet\ (h,j,s,u) \in$ зает$(e)]$
30. $E=\{n \in$ ENSEIGNEMENT $\mid \exists\ a \in$ ANNÉE $\bullet\ (n,a,$истина$) \in$ следва(л) $\}$
    където $t \in$ СТУДЕНТ
31. n = записан $\Rightarrow$ изискван $\subseteq E$

Мрежов модел

Тук основните понятия са : логически запис и връзка. Концептуалната схема се представя като граф, чиито възли за записите и дъгите са връзките. Моделът не поставя ограничения върху концептуалната схема, но повечето СУБД от този тип не позволяват връзки от тип M :N. Мрежовият модел изпитва големи трудности с поддържането на интегритета и сигурността на данните и предлага слаба логическа и физическа независимост.

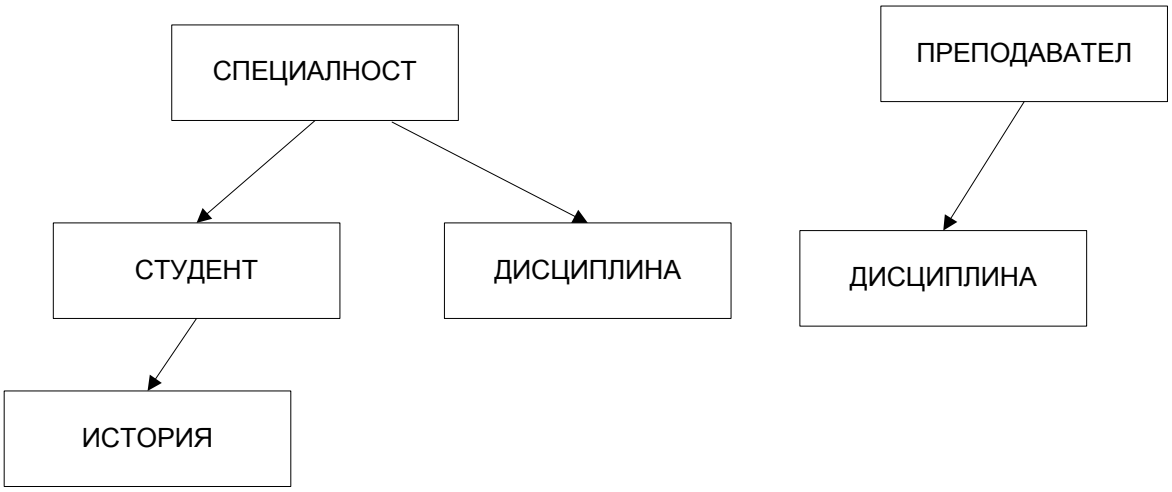


Фигура 2.2 Мрежов модел

Йерархичен модел

Исторически това е първия концептуален модел. Той може да се представи като една или повече дървовидни структури.

В дървото наследникът не може да има повече от един родител (предшественик). Тук отношенията 1:N са възможни само с това ограничение. Моделът има много слаба физическа независимост.



Фигура 2.3 Йерархичен модел

Релационен модел на данните (РМД)

РМД е предложен от E.F. Codd през 1970 г. Той представя множеството на данните като таблици. Всяка таблица преставлява една релация (отношение). Стълбовете на таблицата имат име, което отговаря на всяка от съставните части на релацията. Например, ако трябва да се опишат данните отнасящи се за студентите, дисциплините, преподавателите и записванията в предния пример, трябва да се създадат 4 релации или таблици.

СТУДЕНТ

NO_CT	ИМЕ	АДРЕС
1215	Ана	София
1218	Петър	Пловдив
1230	Иван	Русе

ПРЕПОДАВАТЕЛ

NO_ПРЕП	ИМЕ	ДЛЪЖНОСТ
101	Шишеджиев	Доцент
105	Арнаудов	Професор
110	Ганчев	Доцент

ДИСЦИПЛИНА

NO_ДИСЦ	НАИМЕНОВАНИЕ	NO_ПРЕП
152	Информатика	110
210	Мениджмънт	105
212	Чужд език	101
255	Математика	105

ЗАПИСВАНЕ

NO_CT	NO_ДИСЦ	КУРС	BOOL
1215	152	1996	Взел
1215	210	1996	Скъсан
1215	210	1997	Взел
1218	152	1997	Взел
1230	210	1997	Скъсан

Целите на релационния модел са:

1. Да предложи лесни за използване схеми на данните
   Предимството на този модел е простото понятие за релация или таблица от стойности. При това представяне не се споменава нищо за физическото ниво. Тази простота прави модела използваем за голям брой потребители, които имат по-големи или по-малки познания в информатиката.

2. Намаляване на физическата и логическата зависимости
   Фактът, че моделът не съдържа никакво физическо описание и методите на достъп се извършват независимо от декларациите в логическата схема на релациите, прави възможна физическата независимост. Логическата независимост се гради върху понятието релационен изглед.

3. Предоставяне на потребителите на език от високо ниво
   Езиците за манипулация на данните в СУБД от мрежов тип са от процедурен тип, докато SQL – езикът на РМД е непроцедурен. Той описва каква информация се търси без да описва начина, по който трябва да се получи тази информация. Тази характеристика опростява писането на приложни програми и увеличава физическата независимост.

4. Оптимизиране на достъпа до данните
   Тази цел е пряко следствие от предните две.Тя изисква от СУБД търси най-добрата стратегия за достъп.

5. Подобряване на интегритета и секретността
   Ограниченията за интегритета трябва да се дефинират на концептуално ниво. Само език от четвърто поколение дава тази възможност
   По същата причина само такъв език позволява да се дефинира информацията , която трябва да се засекрети и срещу какво трябва да се засекрети извън пътя и алгоритмите за достъп.

6. Разработка на разнообразни приложения
   Горните характеристики дават възможността приложните програми да се пишат по-бързо и от по-неопитни програмисти.

7. Създаване на методология
   Този модел опростява проектирането, тъй като е по-формализиран и са разработени многобройни методологии за проектиране като E-R модела, MERISE, ORM и други.